PATENT
033401-001

METHOD FOR ADDING METADATA TO DATA

This application claims priority under 35 U.S.C. § 119 to U.S. Provisional Application No. 60/312,788, filed in the U.S. Patent and Trademark Office on 17 August 2001, and which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The invention relates generally to the field of data processing.

2. Background Information

[0002] Currently there are thousands upon thousands of software programs installed in millions of computers that cannot transfer meaning from one to the other. For example, large companies with many branches or subsidiaries often find that the accounting or operating software programs used by one division or subsidiary is not compatible with the software used by other divisions or subsidiaries or the central corporate programs. This requires substantial conversions of data and often results in a great deal of data reentry along with the attendant costs and data integrity problems that attend data entry.

[0003] Because of the great variety of programs, operating systems and software standards currently used by software developers there is a great deal of incompatibility between suppliers and their customers. This also requires substantial conversions of data and often results in a great deal of data reentry and its implications. The unstructured and undefined nature of the current computer software environment imposes great burdens and expense on regulatory organizations such as the SEC, FDIC, Federal and State tax authorities, banks, etc. and the companies reporting to them.

[0004] To overcome this problem many standards organizations have been formed and are being formed to establish defined input/output vocabularies for use with the XML (eXtensible Markup Language) file format. XBRL (eXtensible Business Reporting Language) is one of the XML language formats being

developed. It is expected to become a global standard for financial reporting. Throughout this disclosure we will use XBRL as the example of an XML language. It is not intended to limit the invention to XBRL or XML languages. We find many similarities for the Semantic Web where information Labels are used to facilitate computers talking to computers making decisions and taking action as a result of the communication. Other standards already exist and more will be developed that will benefit by the basic theory of this invention.

[0005] Virtually none of the existing software applications can automatically or semi-automatically convert conventional documents or data into outputs tagged with the standardized Information Labels called for by XML or other standards committees. In most cases the standards themselves are still in development. In order for XML and other data dictionaries or business vocabularies to take root, it is required that existing applications and data be associated or tagged with these standard vocabularies. This harsh reality will long delay the widespread use of these standards because it will take years for companies to migrate to new software products that are designed to output the appropriate Information Labels. In some cases that may never happen because it is virtually impossible to replace legacy software systems. For example, retrofitting all the accounting software in current use would be a very complex task that could not be accomplished in any short-term.

[0006] The recognized practical approach to standardizing the meaning of data is to attach defined Information Labels to the information being conveyed. In this way the meaning of the data can be determined by reviewing the definition of the label. It also means that computers can recognize the "meaning" of the tagged information and act on it based on that meaning. For example, data with the same "tag" can be added or compared without fear of adding or comparing apples and oranges.

[0007] Taxonomies and their extensions are used to define the Information Labels. For example in a financial report, the label <Sales> followed by a numerical value indicates that the numerical value relates to company's Sales. <Cost of Goods

Sold> followed by a numerical value indicates that the value represents the company's Cost of Goods Sold. Since Gross Profit is Sales minus Cost of Goods Sold, computers could access 3rd party reports that show these values and easily calculate the Gross Profit with a simple rule that says <Sales> <minus> <Cost of Goods Sold> = <Gross Profit>.

[0008] Because not all companies use the same terminology, the taxonomies used by standards organizations also include synonyms and alternative phrases that have the same meaning. For example synonyms for Sales could include "Revenues" or "Fees". Cost of Goods Sold might be "Cost of Goods" or "Cost of Sales". The Information Labels can also carry information regarding the organizational authority that defined the label. If the taxonomy were authored by the US Securities & Exchange Commission the labels based on that taxonomy might be identified as USSEC, and so on.

[0009] Accordingly, there is a need for methods and mechanisms to accurately and efficiently transform data into XML, and in particular XBRL, compliant formats. The transformation would include, for example, adding appropriate labels to the data as defined in relevant XBRL taxonomies. There is also a need for methods and mechanisms to automate entry of XML and XBRL compliant data into non-XML or non-XBRL compliant programs or applications.

[0010] *XBRL Essentials*, authored by Charles Hoffman and Carolyn Strand, copyright 2001 by XBRL Solutions, Inc., ISBN 0-87051-353-2, is hereby incorporated by reference.

SUMMARY OF THE INVENTION

[0011] In an exemplary embodiment of the invention, a data stream is captured, data in the captured stream are identified, and then the identified data are mapped to a file structure, a schema, or a taxonomy. In exemplary embodiments of the invention, the output data stream is a data stream to a display screen, a memory, a hard drive, a CD ROM drive, a floppy disk drive, or a printer. The output data stream can be conveyed through serial or parallel ports (including

Universal Serial Bus or "USB", FireWire™,), via wireless interfaces, and so forth. In other exemplary embodiments of the invention, the identified data are mapped to an XBRL (eXtensible Business Reporting Language) taxonomy, a spreadsheet, a database, or a flat file.

[0012] In another exemplary embodiment of the present invention, a method for adding labels to data includes a) identifying data in an electronically represented file, b) selecting labels that correspond to text strings in the identified data, based on a list associating labels with text strings, and c) adding the selected labels into the electronically represented file to label the text strings and elements in the identified data associated with the text strings. The labels include information about the data and are defined in one or more taxonomies. In the event the list does not associate a label with the text string, a user can be prompted to select a label corresponding to a text string in the identified data. The association indicated by the user's selection, can then be added to the list associating labels with text strings. Preferably the labels are consistent with XML (eXtensible Markup Language), and also conform to an XBRL (eXtensible Business Reporting Language) specification. This embodiment can be implemented by a transformation program that receives the electronically represented file from a target program. The transformation program a) performs the steps of identifying, selecting and adding, and b) is configured to appear to the target program as a printer driver. The transformation program can be independent and separate from the target program.

[0013] In accordance with another embodiment of the invention, a method is provided for importing at least a portion of an XBRL compliant data set into a non XBRL compliant target application. The method includes the steps of exporting data from the target program in an export file, a user associating entries in the export file with labels defined in one or more appropriate XBRL taxonomies, and forming an import file for import into the target program by replacing data in the export file at entries associated with specific labels, with data

from the data set having corresponding labels. The associations made by the user are stored for later use, so that an import file can be automatically created by replacing data in a file having the same format as the originally exported file, based on the stored associations.

[0014] In accordance with another embodiment of the invention, a method is provided for importing at least a portion of a set of data into a target application, where the data set including labels indicating information about data in the data set, and where the labels are defined in one or more taxonomies. For example, where the data set is XBRL compliant and the labels are defined in one or more XBRL taxonomies. The method includes a data entry program observing a user entering data associated with the labels into the target application, and storing key strokes associated with the entry of data for each different label. Then, when the data entry program receives an XBRL compliant data set for entry into the target application (which can be non XBRL and non XML compliant), the data entry program can enter the data from the data set into the target application, by performing the stored key strokes corresponding to the labels associated with the data in the data set. When the data entry program is automatically entering data into the target application, and encounters a data item having a label for which no keystrokes are stored, the data entry program can prompt the user to enter the data item into the target application, and then observe and store the user's keystrokes for future use.

[0015] In accordance with another embodiment of the invention, a method is provided for importing at least a portion of a data set into a target database. The method includes entering test data into the target database, and then searching or scanning the database for patterns corresponding to the test data. A pattern recognition application that is independent from the database can be used for this purpose. A structure of the database is modeled based on the search results. Thereafter, the database can be directly accessed using the modeled structure. In particular, the modeling process includes associating locations within the database

structure with labels, where the labels correspond to elements of the test data that were found at the locations during the step of searching. A data element can then be imported directly to a specific location within the database, using for example an independent software application, based on a label associated with both the location and the element.

[0016] Exemplary embodiments of the invention include a synonym dictionary that includes synonyms of known labels or terms, or synonymous links between labels and/or terms, to facilitate automatic or user-assisted mapping. The dictionary can include terms that are not part of a taxonomy or schema such as an XML taxonomy, but that are synonymously related to terms in a taxonomy, schema, *etc.* In an exemplary embodiment of the invention, the synonym dictionary includes foreign languages, so that a label or datum can be mapped from one language into another language. In an exemplary embodiment of the invention, currency values are identified in the data stream, and are converted to corresponding values in different currencies (*e.g.,* from yen to dollars) based on a known or designated exchange rate. In accordance with an embodiment of the invention, the mapping process converts data from one standard to another, for example from U.S. GAAP (Generally Accepted Accounting Principles) to International GAAP. In accordance with an embodiment of the invention, the mapping process includes replacing labels corresponding to identified data, with other labels, for example where minimizing file size is important.

[0017] In accordance with an embodiment of the invention, data output from a first computer platform or system can be automatically converted by a software module on the first platform, from a first format into an intermediate format, transferred to a second platform or system, and then converted from the intermediate format into a second format by a second software module on the second platform. For example, the intermediate format can be an XML taxonomy, and the software modules can effectively "translate" so that data can be transparently exchanged between the two platforms regardless of whether the first

and second formats are compatible or known to each of the two platforms. The intermediate format can also be encrypted, *e.g.* for secure transfer.

[0018] In accordance with embodiments of the invention, the processing steps and mechanisms described above, are performed in a remote or distributed fashion, in realtime or non-realtime.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] Other objects and advantages of the present invention will become apparent to those skilled in the art from the following detailed description of preferred embodiments, when read in conjunction with the accompanying drawings wherein like elements have been designated with like reference numerals and wherein:

[0020] FIG. 1A shows a flowchart in accordance with an exemplary embodiment of the invention.

[0021] FIG. 2A shows a flowchart in accordance with an exemplary embodiment of the invention.

[0022] FIG. 2 shows a flowchart in accordance with an exemplary embodiment of the invention.

[0023] FIG. 3 shows a flowchart in accordance with an exemplary embodiment of the invention.

[0024] FIG. 4 shows a flowchart in accordance with an exemplary embodiment of the invention.

[0025] FIG. 5 shows a relationship between a target program and a transformation program in accordance with an embodiment of the invention.

[0026] FIG. 6 shows a relationship between a target module and a transformation module in accordance with an embodiment of the invention.

[0027] FIG. 7 shows software layers in an exemplary embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0028] In accordance with an embodiment of the invention shown in FIG. 1A, a data stream is captured in step 150, data in the captured data stream are identified in step 152, and then in step 154 the identified data are mapped to a file structure, a schema, or a taxonomy. The output data stream is a data stream to a display screen, a memory, a hard drive, a CD ROM drive, a floppy disk drive, or a printer. The output data stream can be conveyed within a computer, through serial or parallel ports (including Universal Serial Bus or "USB", FireWire™, *etc.*), via wireless interfaces, and so forth, and can be captured via duplication or redirection, at any point along the conveyance, via software and/or hardware mechanisms. The identified data are mapped to an XBRL (eXtensible Business Reporting Language) taxonomy, a spreadsheet, a database, an XML (eXtensible Markup Language) taxonomy, a standard (*e.g.*, U.S. GAAP or International GAAP), or a flat file. When the identified data are mapped to a flat file, a specification or "data definition" file can also be generated to indicate the meaning or character of information at different locations in the flat file (*e.g.*, in different columns, at different locations within a given text string, *etc.*), and to optionally indicate delimiters (*e.g.* tabs, commas, spaces, semicolons, *etc.*) between discrete elements of information or groups of information in the flat file. The flat file and an accompanying data definition can, for example, be generated in accordance with known techniques and formats relating to flat files.

[0029] The embodiment shown in FIG. 1A, can be implemented as shown in FIG. 1B. In accordance with an exemplary embodiment of the invention illustrated in FIG. 1B, a method for adding labels to data includes a) identifying data in an electronically represented file, b) selecting labels that correspond to metadata in the identified data, based on a list associating labels with metadata, and c) adding the selected labels into the electronically represented file to label the metadata and/or elements in the identified data associated with the metadata. The labels include information about the data and are defined in one or more taxonomies. In the context of the present application, "metadata" or "meta

information" is data about data, or information that describes other information. In this example the metadata in the identified data identifies or describes other data elements within the identified data, and can include for example text strings, various control characters (*e.g.*, various ASCII control characters), and so forth For example, metadata in the captured data stream or file can be used to identify the data to which the metadata refer, and then additional metadata referring to the identified data can be added to the captured data stream or file. For example, the list can contain labels from multiple taxonomies, standards, and so forth, including words from languages, link synonymous or related labels. When a label from a first taxonomy, *etc.* is recognized in the captured data stream or file, the data element it labels can also be further labeled with a corresponding label from a second, different taxonomy, standard, *etc.* Thus a computer program that recognizes the second taxonomy but not the first, will now be able to use or recognize and organize the information in the data stream or file. A new, transformed data stream or file can be formed by adding the new labels for the second taxonomy, and optionally removing the old labels from the first taxonomy (or standard, schema, *etc.*).

[0030] In the event the list does not associate a label with metadata in the identified data, a user can be prompted to select a label corresponding to the metadata. The association indicated by the user's selection, can then be added to the list associating labels with metadata. Preferably the labels are consistent with XML (eXtensible Markup Language), and also conform to an XBRL (eXtensible Business Reporting Language) specification. Of course, the labels can also be consistent with data formats for spreadsheets, relational databases, and other file structures or schemas or standards.

[0031] This embodiment can be implemented by a transformation program that receives the electronically represented file from a target program. The transformation program a) performs the steps of identifying, selecting and adding, and b) can be configured to appear to the target program as a type of software

known to the target program. For example, the transformation program can appear to the target program as a printer driver.

[0032] The transformation program can be independent and separate from the target program. The transformation program can also be entirely resident on the same computer or system as the target program, or can be remotely located on a different system, or distributed among different systems. The transformation module can be a single module, or a plurality of cooperating modules. A list and/or synonym dictionary that the transformation program or module(s) use to identify metadata and add corresponding metadata, can be stored as a data file separately from the program or module(s), and can be stored or accessed remotely, for example via an Internet web server.

[0033] For example, the data stream can be captured at an information provider's site, transferred (as a real-time stream of data or as a data file containing contents of the captured data stream) to another location such as an intermediate location or the information receiver's site, and then provided to the information receiver's site. The information provider computer could have, for example, a transformation program emulating a print driver, that is selected when information is to be output for mapping. The output would be provided to the transformation program, and then conveyed to the information receiver machine (by email, modem, file on floppy disk, *etc.*). A transformation program on the information receiver machine would then open or receive the data, and map it to a batch file format useable by a target import program or to a file format useable by a program written to update a database.

[0034] The transformation programs on the provider and receiver machines can be identical and both capable of receiving, transferring and mapping data, or can have different capabilities. For example, the transformation programs can be configured to handle an intermediate format so that the transformation program at the information provider would map the data to an intermediate format, and transfer the data in the intermediate format to the transformation program on the

receiver machine. The receiver machine would map the data from the intermediate format to another format useful on the receiver machine (or as desired by a user). The programs could be different versions, so that the transformation program recognizes more formats than the transformation program at the receiver machine and thus can map more formats to or from the intermediate format. In addition or as an alternative, the transformation program on the receiver machine can be configured or featured to only map the data out of the intermediate format to another format, without being able to map data into the intermediate format in much the same way that Adobe Acrobat™ Readers can open and view, but not create, .pdf files. The transformation programs can also be configured to operate automatically without user intervention. For example, the transformation program on the provider machine can automatically transfer data in response to a request from the transformation program on the receiver machine, subject for example to rules or requirements (*e.g.*, a user's prior approval to allow public access to information on the provider machine) in place on the provider machine. The provider and receiver machines can communicate via the Internet. For example, the provider machine can interface the Internet or function as a web server, and the receiver machine can interface the Internet or function as a web browser. Also, the intermediate format can be encrypted, and can be decrypted at the receiver machine in a fashion transparent to a user of the receiver machine. For example, the encryption/decryption mechanism can be a proprietary function of the transformation programs.

[0035] The transformation program can alter or transform the file it receives from the target program, for example by adding appropriate XBRL labels to the file. Alternatively, the transformation program can combine data from the file received from the target program, with the selected labels to generate and output a new, transformed file. As a further alternative, the transformation program can replace labels in the file with the newly added labels, for example when converting from one standard or language to another. This is advantageous

when it is desirable to minimize the size and complexity of the transformed file or transformed data stream.

[0036] As shown in FIG. 1B, in a first step 102, data in an electronically represented file is identified. Next, in step 104, labels are selected that correspond to metadata such as text strings in the identified data, based on a list that associates labels with text strings. Although "text strings" are specifically referred to in FIG. 1B, "metadata" can be substituted for each occurrence of "text string(s)". In other words, the concepts shown in FIG. 1B apply also to all other forms of metadata, not just to text strings. This also holds true for the other embodiments described herein.

[0037] From step 104 control proceeds to step 106, where a determination is made whether an un-identified text string, or a text string that does not have an associated label on the list, has been encountered. If yes, then control proceeds to step 108, where the user is prompted to select a label that corresponds to the text string. For example, the user can be provided with one or more taxonomies in a pop-up window or as part of the dialog, so that the correct label can be quickly and easily selected.

[0038] From step 108, control proceeds to step 110. In step 110, an association selected by the user in response to the prompt is stored for future use. From step 110, control proceeds to step 112. If in step 106 the determination is negative, then control proceeds from step 106 to step 112.

[0039] In step 112, a determination is made whether labels have been selected (using the list, for example) for all relevant text strings in the identified data. The assumption here is that there will be a label in some form associated with each datum, which can be used to map the datum to an appropriate label in, for example, an XBRL taxonomy. The software application performing this function can exercise a degree of intelligence to filter out extraneous or superfluous text, and to properly interpret text and nearby data. For example, in the output from an accounting system, say a Balance Sheet, the output may contain a Report Header

and a Report Footer, one or both of which need not be translated depending on the circumstances. Also, it is possible that the text being interpreted and correlated with an XBRL label, may span more than one line but data related to the text will be only on one line. In this situation the software application would appropriately merge multiple lines. In addition, it is possible that a text string may be a label referring or applying to multiple items of data, for example a financial statement with a text label called "cash on hand" and another label for the reporting period of "2000". Placement or location of a datum in the file can also help indicate which XBRL label is appropriate for the datum. Any information relative to the position of the datum in relationship to other data that helps to label it (for example, a placement in a document that would show a data item nested in a specific location within another item, like a hierarchy), can be used help determine an appropriate XBRL label for the datum.

[0040] If in step 112 the determination is negative, then control returns to step 104. If in step 112 the determination is positive, then control proceeds from step 112 to step 114. In step 114, the data are re-formatted in accordance with selected labels. In other words, the data are re-formatted based on the determined correspondence between the data and defined labels in one or more XBRL taxonomies. This re-formatting can include adding the corresponding XBRL labels into the data. As indicated in step 116, the reformatting can also include re-ordering the data in accordance with a hierarchy of the selected/corresponding XBRL labels.

[0041] In summary, the transformation program can transform the data in various ways, including inserting and/or interpreting information labels or tags used to describe, characterize, and/or organize the data, to make the data more usable. The transformation program can be made appropriately compatible with various operating systems, including (but not limited to) MS Windows, Unix, Mac OS, Solaris, Linux, and so forth. The transformation program can acquire the data file to be transformed in any of various formats, including as a database file, a flat

file, EDI, screen data, or any other collection or stream of data that can be analyzed in a digital format. The transformation program can also output a transformation file including the transformed data, in any appropriate format. For example, the output file can be in any format that is XBRL compliant.

[0042] The transformation program can also launch or invoke an application or submodule to validate the output file, and can launch a Compare Program to analyze a received file by comparing text strings in the File with a standardized XBRL taxonomy. Then, the transformation program can compare the text strings in the file with the appropriate XBRL taxonomy (including Synonyms). The comparison may be done either by parsing the data or by using Rev-Gen pattern recognition scanning techniques. Any previous User mapping of XBRL Information Labels to data can also be checked.

[0043] The transformation program can also link the appropriate XBRL Information Label to the related information whenever such a link can be clearly established without user intervention. Any text strings that cannot be automatically identified and linked with XBRL taxonomy Information Label will be presented to the User on the first occurrence. Using drag and drop or any other convenient mapping technique, the user will link the information in question with the appropriate XBRL Information Label (tag).

[0044] For example, the first time the company publishes financial statements using this technique the name of the company may not be recognized as <Company Name> data. To link the <Company Name> label with the company name data, the user would simply drag the <Company Name> Information Label to the name of the company and the link would be established. This link would then remain in the Transformation Program for subsequent reports so the User would make this connection only once.

[0045] The transformation program also can create a new XBRL output file that includes all the appropriate Information Labels, Style information and the proper XML file extension to be XBRL compliant. Once the XBRL Information

Labels have been linked to the appropriate data, some of the steps can be bypassed when producing subsequent reports unless a term in the application program has been changed or a new term has been added to the report.

[0046] Exemplary embodiments of the invention include a synonym dictionary that includes synonyms of known labels or terms, or synonymous links between labels and/or terms, to facilitate automatic or user-assisted mapping. For example, where a known label in a standard, schema or taxonomy to which captured data stream or file is being mapped is "Sales", the dictionary can include synonyms such as "Fees" and "Revenues" so that when the synonyms are identified in the captured data stream the datum they refer to will be mapped appropriately to (or labeled with) the label "Sales". The synonym dictionary can be incorporated within the list associating data and metadata. The dictionary can include terms that are not part of a taxonomy or schema such as an XML taxonomy, but that are synonymously related to terms in a taxonomy, schema, *etc.* In an exemplary embodiment of the invention, the synonym dictionary includes foreign languages, so that a label or datum can be mapped from one language into another language.

[0047] For example, the transformation program can also be used to translate terms in a document from one language to another. For example, the list associating data and metadata, which the transformation program uses to identify data and select additional or replacement labels, can include languages or portions of languages together links indicating synonyms among the languages. The language portions can be, for example, English language descriptive terms that appear in the U.S. GAAP, and corresponding synonyms in French, German, Spanish, *etc.*, and similar terms that might appear in other standards such as International GAAP. Thus, a user can provide a document containing financial information consistent with U.S. GAAP, to the transformation program, and specify that the transformation program output the document with French words instead of English words. A user can also request the transformation program to

convert the U.S. GAAP document into an International GAAP document with German words instead of English words, and so forth. The user can specify the desired output language, and optionally the original language. The transformation program can automatically identify the original language, for example when it finds labels in the captured data, that correspond to labels in its list, that it knows are in a specific language.

[0048] In addition, in an exemplary embodiment of the invention, the transformation program can be used to identify currency values identified in the captured data stream or file, and then convert the identified currency values to corresponding values in different currencies (*e.g.,* from yen to dollars) based on a known or designated exchange rate. A default exchange rate can be used, for example the exchange rate that was in effect when a) the original data were created, b) the data stream or file was captured, c) the conversion was performed, or d) a date indicated by a user. The user can also specify the exchange rate.

[0049] In accordance with another embodiment of the invention illustrated in FIG. 2, a method is provided for importing at least a portion of an XBRL compliant data set into a non XBRL compliant target application. The method includes the steps of exporting data from the target program in an export file, a user associating entries in the export file with labels defined in one or more appropriate XBRL taxonomies, and forming an import file for import into the target program by replacing data in the export file at entries associated with specific labels, with data from the data set having corresponding labels. The associations made by the user are stored for later use, so that an import file can be automatically created by replacing data in a file having the same format as the originally exported file, based on the stored associations. An import file template can be generated based on the structure of the export file and the associations made by the user, and an import file can then be formed by populating the import file template with data by entering the data based on labels associated with both the data being entered and entries in the import file template. The template can of

course be reused to import different sets of data. The user can indicate associations between entries in an export/import file format in any appropriate or suitable way. For example, the user can insert data associated with labels into various entries of the export file, and then software can scan the entries in the export file, discern the associated labels based on the newly entered data, and then store the associations for later use when populating an (empty) import file template with data for import into the target program or target application. A structure of the export file together with the stored associations can represent an import file template. The newly entered data can include the labels themselves. Alternatively, software can, for each entry in the export file, present a list of labels, and a user can select one or more appropriate labels from the list to indicate the association, which is then stored. The template can be populated with data for import, for example, by discerning a label associated with a datum to be imported, locating an entry in the template associated with the same label, entering the datum into the located entry in the template, and repeating these steps for all data in a data set to be imported.

[0050] As shown in FIG. 2, in step 202 data is exported from target application or program in an export file. From step 202 control proceeds to step 204, where a user associates entries in the export file, with labels, for example labels defined in an XBRL taxonomy. From step 204 control proceeds to step 206, where the associations made by the user are stored. From step 206 control proceeds to step 208, where an import file is generated by replacing data in the export file at entries or locations associated with the (*e.g.*, XBRL) labels, with new data having corresponding labels.

[0051] In another embodiment of the invention illustrated in FIG. 3, a method is provided for importing at least a portion of a set of data into a target application, where the data set including labels indicating information about data in the data set, and where the labels are defined in one or more taxonomies. For example, where the data set is XBRL compliant and the labels are defined in one or more XBRL taxonomies. The method includes a program observing a user

entering data associated with the labels into the target application, and storing key strokes associated with the entry of data for each different label. Then, when the data entry program receives an XBRL compliant data set for entry into the target application (which can be non XBRL and non XML compliant), this program or a different program can enter the data from the data set into the target application, by performing the stored key strokes corresponding to the labels associated with the data in the data set. If the program that is automatically entering data into the target application, encounters a data item having a label for which no keystrokes are stored, it can prompt the user to enter the data item into the target application, and then observe and store the user's keystrokes for future use.

[0052] As shown in FIG. 3, in a first step 302 a first software application observes a user entering data associated with the labels, into a target application. From step 302 control proceeds to step 304, wherein the first application stores observed keystrokes associated with entry of data for each different label (*e.g.*, XBRL label). From step 304, control proceeds to step 306, where the first application receives a data set for entry into the target application. From step 306, control proceeds to the 308, where the first application enters data from the data set into the target application, by performing the stored keystrokes corresponding to the labels associated with the data in the data set. From step 308, control proceeds to step 310, where a determination is made by the first application, whether it has encountered any data in the data set for which it has no stored keystrokes. In other words, whether there is any data in the data set having a label for which the first application has not stored or observed any keystrokes. If yes, then control proceeds to step 312, where the first application prompts the user to enter the data item into the target application, or otherwise provide an appropriate sequence of keystrokes to enter the data item into the target application. For example, an appropriate sequence could be selected from a menu or group of pre-recorded keystroke sequences. From step 312, control proceeds to step 314, where

the provided keystroke sequence is stored for future use by the first application. From step 314, control proceeds to step 316.

[0053] If in step 310 the determination is negative, then control proceeds to step 316.

[0054] In step 316, the first application determines whether all relevant data in the data set has been entered into the target application. If yes, then control proceeds to step 318, where the process ends. If no, then control returns to step 308. "Relevant" data can be determined or handled subject to the considerations discussed above with respect to step 112 of FIG. 1.

[0055] In accordance with another embodiment of the invention illustrated in FIG. 4, a method is provided for importing or inputting at least a portion of a data set into a target database. The method includes entering test data into the target database, and then searching or scanning the database for patterns corresponding to the test data. A pattern recognition application that is independent from the database can be used for this purpose. A structure of the database is modeled based on the search results. Thereafter, the database can be directly accessed using the modeled structure. In particular, the modeling process includes associating locations within the database structure with labels, where the labels correspond to elements of the test data that were found at the locations during the step of searching. A data element can then be inserted directly to a specific location within the database, using for example an independent software application, based on a label associated with both the location and the element.

[0056] As shown in FIG. 4, in a step 402, a set of test data is imported or inputted into a target database. Preferably the set is entered into the database in a conventional fashion, for example by key entry through an interface of an application that manages the database. The database can be separate from the managing application, or can be embedded within the managing application. From step 402, control proceeds to step 404, where the database is scanned by an independent software application, for example a pattern recognition application

such as that manufactured by the British company RevGen Plc. and distributed by their U.S. affiliate, Generos corporation. The independent application searches or scans the database for patterns corresponding to the set of test data.

[0057] From step 404 control proceeds to step 406, where an independent application (for example, the pattern recognition application or another, separate application) constructs a model of the structure of the database, based on the search/scan results. From step 406 control proceeds to step 408, where locations in the database structure are associated with labels, for example labels defined in one or more XBRL taxonomies. The labels correspond to elements of the test data found at those locations in the database structure during the search/scan. From step 408, control proceeds to step 410, where an element from a data set is imported directed into the database based on a label associated with both the location and the element.

[0058] FIG. 5 shows a transformation program consistent with the embodiment described in FIG. 1. As shown in FIG. 5, the transformation program can be independent and separate from the target program. Specifically, a target program 502 provides an output file such as a print file to a transformation program 504. The transformation program 504 is configured so that it appears to the target program as a printer driver. The transformation program 504 does not require or perform any modification or alteration of the target program 502, and can be designed or configured to function compatibly with commercially available target programs such as spreadsheets, accounting programs, word processing programs, and so forth.

[0059] FIG. 6 shows that the transformation program and the target program of the embodiment described in FIG. 1, can alternatively be implemented respectively as a module 604 and a target program module 602 together within an application 606. The module 604 can appear as a printer driver to the module 602. For example, the transformation program module 604 can be implemented as a

DLL, OCX, Active X Control program, or in any other form that can be marketed to software vendors for integration into independently developed applications.

[0060] FIG. 7 shows an exemplary structure of various embodiments of the invention, with respect to the software layers of a computer. In particular, FIG. 7 shows an application layer 702, at which any Windows™ application 701 operates on the computer. Below the layer 702 is a Windows™ OS (Operating System) layer 704, in which can be found a GDI (Graphics Interface Device) 703. Below the layer 704 is a low level OS interface layer 706, at which an XBRL Printer Driver 705 in accordance with the invention can be found. Below the layer 706, is an XML mapping agent layer 708, with an XML Mapping Agent 707. Below the layer 708 is a data conversion layer 710, which includes a Data Converter 709 that outputs or can output the data in a variety of formats, including the formats 712-722 shown (HTML, Excel™, XML, SQL, xBase, and ASCII respectively). In accordance with various exemplary embodiments of the invention, a transformation program that performs various functions of the invention includes the XBRL Printer Driver 705, the XBRL Mapping Agent 707, and the Data Converter 709. Although the elements 705, 707 are shown as being XBRL-related, the elements 705, 707 can be related to any or all of the formats, taxonomies, protocols, standards, etc. described above and their equivalents. In addition, the formats 712-722 are exemplary and not limiting.

[0061] With respect to each of the described embodiments, information provided by the user, for example associations between data from a target application and XBRL labels or tags, can be made using drag-and-drop, cut-and-paste, selection of items from a proffered menu, keyboard entry, or any other appropriate technique. In addition, the described embodiments can be variously combined. Extracting data from the target program or target application can include, in addition to or instead of obtaining a print file, accessing data directly from a file or out of a database without running or launching the parent (target) application, scraping data off of a display screen or window, and so forth.

[0062] Those skilled in the art will recognize that the software functions described herein can be variously implemented as a) software instructions running on a hardware machine such as a desktop computer having a central microprocessor, b) appropriately configured Field Programmable Gate Array(s) (FPGAs), c) Application Specific Integrated Circuit(s) (ASICs), or any other equivalent or suitable computation device.

[0063] It will be appreciated by those skilled in the art that the present invention can be embodied in other specific forms without departing from the spirit or essential characteristics thereof, and that the invention is not limited to the specific embodiments described herein. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes that come within the meaning and range and equivalents thereof are intended to be embraced therein.